MULTIMEDIA UNIVERSITY

STUDENT ID NO | | | | | | | | | | |

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2022/2023 (TERM ID 2210 & TERM ID 2215)

DOO5441 – OBJECT ORIENTED PROGRAMMING

(For DIT students only)

14 FEBRUARY 2023
2.30 P.M – 4.30 P.M
(2 Hours)

INSTRUCTION TO STUDENTS

1. This examination paper consists of 7 pages.
2. **SECTION A:** There are **2** structured questions.
3. **SECTION B:** There is **1** structured question.

SECTION A: Structured Questions (Total: 20 Marks)

Instruction: Please write all your answers in the Answer Booklet provided.

QUESTION 1 (10 marks)

A. Given the following code segments;

i. Identify the final value of k.

```
    int w, g, k;
    w = 5;
    g = 8;
    while ( g > w )
    {
      g = 2 * w++;
      w = g * ++w;
    }
k = w;
```

[1 mark]

B. Write the output when the following code is executed.

```
int k = 5, t = 9;
char *ptr;
ptr = "Multimedia University";
if ( t < 5)
 cout << k+t << "\t" << *(ptr + k);
else
 cout << k*t << "\t" << *(ptr + t);
```

[1 mark]

C. Write program segments based on the descriptions given:

i. Create a structure called *Subject.* The data members are:

- *Name* ***(string)***
- *ID* ***(string)***
- *LecGrp* ***(string)***
- *Final****(double)***
- Grade***(string)***

[3 marks]

ii. Declare an object named *OOP* with size 3 of struct *Subject.* [1 mark]

Continued...

D. Given an incomplete code below.

```
int main()
 {
   string pet;
   char spayed;
   float fees;

   cout << "Enter the pet type (cat or dog): ";
   cin  >> pet;
   cout << "Has the pet been spayed or neutered (y/n)? ";
   cin  >> spayed;

   -----(a)---------

   cout<< "The cost is RM"<<fees;
   return 0;
 }
```

Write an *if-else statement* at ***–(a)–*** to determine the *fees* based on the table given.

[4 marks]

Pet	Spayed Status	
	Spayed = Yes	Spayed = No
Cat	fees = RM50.00	fees = RM230.00
Dog	fees = RM200.00	fees = RM350.00

Continued...

QUESTION 2 (10 Marks)

Given the program shown below, answer the following questions:

```
#include <iostream>                                                    //Line 1
##include <cmath>                                                     //Line 2
using namespace std;                                                   //Line 3
class Sphere                                                           //Line 4
{     float radius, volume, PI=3.142;                                  //Line 5
      private:                                                         //Line 6
       void Sphere(float r)                                            //Line 7
          {   radius = r + 1;}                                         //Line 8
       Sphere()                                                        //Line 9
          {  radius = 0.5 + 1;}                                       //Line 10
            void setData(float r)                                     //Line 11
          { radius = r; }                                             //Line 12
                                                                      //Line 13
          float CalculateVolume()                                     //Line 14
         { volume = (4 * PI * pow(radius,3))/3;                       //Line 15
             return volume;  }                                        //Line 16
} ;                                                                   //Line 17
                                                                      //Line 18
int main()                                                            //Line 19
{                                                                     //Line 20
      Sphere S1, S2,S4(1.8),S3;                                       //Line 21
      float r=3.3;                                                    //Line 22
      S1.setData(r);                                                  //Line 23
      S2.setData(5.2);                                                //Line 24
      S3 = S1 ;                                                       //Line 25
cout<<"Volume of Sphere S1 :"<<S1.CalculateVolume()<<endl;            //Line 26
cout<<"Volume of Sphere S2 :"<<CalculateVolume() <<endl;              //Line 27
cout<<"Volume of Sphere S3 :"<<S3.CalculateVolume()<<endl;            //Line 28
cout<<"Volume of Sphere S4 :"<< S4.CalculateVolume()<< endl;          //Line29

return 0;                                                             //Line 30
}                                                                     //Line 31
```

i. Identify **FOUR** lines that contain errors. Fix the errors by writing down the correct answer. [2 marks]

ii. Write the output of the program if the correction has been made. [2 marks]

iii. Write the statements to swap the contents between ***S2*** and ***S4***. [2 marks]

iv. In main (), write the code to write all S1, S2, and S4 volume results into a file named "SphereVolume.txt". [4 marks]

Continued...

SECTION B (Total: 20 Marks)

Instruction: Please write all your answers in the Answer Booklet provided.

Pena Book Company is a publishing company that sells non-fiction books. Pena Book Company would like to create a system to calculate the bonus and net salary details of their staff. There are three different types of staff they are hiring: permanent, contract, and part-time staff.

Write a complete program that gets input from an admin to calculate bonus and net salary for Pena Book Company staff. Three derived classes and one base class will be in the program. The program will display the summary of the staff name, bonus, and salary details.

[Note: Refer to sample output given below. The bold items are inputs entered by admin.]

- Include three pre-processor headers which are iostream, cstring and iomanip to support your codes for this program.
- Create class called **Pena_Book**:
  - Protected data members:
    - *name* : string
    - *THour, Msalary, net_salary, bonus* : double
  - Public member functions:
    - Create a default constructor that displays the message "*Welcome to Pena Book Salary System*". Initialize *name* to the value "*user*".
    - Create a destructor that displays the message "*Thank You.*".
    - Define a virtual function called *set_data(...)* that has parameter *n* (string) and *hour* (double). Initialize the data members' *n* and *hour* to *name* and *THour* accordingly.
    - Define **FOUR** accessor functions for this class.
      - *getname()*: return data member name.
      - *getbonus()* : return data member *bonus*.
      - *getsalary()*: return data member *Msalary*.
      - *getnetsalary()* : return data member *net_salary*.
    - Define a function called *Display_Summary()* that display *name, bonus, Msalary* and *net_salary*.

- Create a derived class called **Permanent_hour** from the base class **Pena_Book** *[use public derivation]*
  - Private data member:
    - *salary = 1600, transportation = 0.0* : double
    - *meals = 0.0, percent=0* : double
  - Public member functions:

Continued...

- Define a function called *set_data(...)* that has a parameter *n* (string) and initialize data member *name* by the parameter *n.*
- Define function called *set_other(...)* that has a parameter *tport* (double) and *food* (double). Initialize the data *tport* to *transportation* allowence and *food* to *meals* allowance.
- Define a function called *cal_net_salary()* and prompt user to enter the bonus percent. Set *Msalary* with salary. Calculate *bonus* and *net_salary* using the following formula:

bonus = salary * percent/100
net_salary = salary + bonus + transportation + meals

➢ Create a derived class called **Contract_hour** from the base class **Pena_Book**. *[use public derivation]*

- Protected data member:
  - *Salary, pay =0, percent=0;* : double
- Public member functions:
  - Define a function called *set_data(...)* that has parameter *n* (string) and *hour* (double). Initialize the *n* and *hour* to data members *name* and T*hour* accordingly.
  - Define a function called *cal_net_salary()*. Prompt user to get the bonus percentage and payment per hour. Calculate *salary* and set *Msalary* to *salary*. Calculate the *bonus*, and *net_salary* using the following formula:

salary = Thour * pay
bonus = salary * (percent /100)
net_salary = salary + bonus

➢ Create a derived class **Part_timer_hour** from the class **contract_hour**. *[use public derivation]*

- Public member functions:
  - Define a function called *repeat()* that called function *set_data(...) and cal_net_salary()* from the class **contract_hour** (using :: operator) .

In ***main()*:**

- Declare three variables which are *name*(string), *type*(string), *meal*(double), *transport*(double), and *hour*(double).
- Using ***for loop*** statement, repeat the process for 3 times:
  - Prompt user to enter *name* and *type* of work (Permanent or Contract or Part Time).
  - If *type* is equal to "Permanent":
    - Use the pointer object called *ph* to assign by a new dynamic object of **Permanent_hour** class.
    - Call function *set_data(...)* and pass parameter *name* by the object *ph.*

Continued...

- Prompt user to enter transportation allowance and meal allowance.
- Call function set_other(...) and pass parameter *transport* and *meal* by the object *ph*.
- Call function *cal_net_salary()* by the object *ph*.
- Display *name, bonus, salary* and *net_salary* for normal hour worker by calling the accessor functions.

- If type is equal to "Contract":
  - Use the pointer object called *ch* to assign by a new dynamic object of **contract_hour** class.
  - Prompt user to enter working hour.
  - Call function *set_data(...)*, pass parameter *name* and *hour* by the object *ch*.
  - Call function *cal_net_salary()* by the object *ch*.
  - Call function *Display_Summary* by the object *ch*.

- If type is equal to "Part-Time":
  - Use the pointer object called *pth* to point to assign by a new dynamic object of **part_timer_Hour** class.
  - Prompt user to enter working *hour*.
  - Call function *set_data(...)*, pass parameter *name* and *hour* by the object *pth*.
  - Call function *cal_net_salary()* by the object *pth*.
  - Call function *Display_Summary()* by the object *pth*.
- Others:
  - Create an object called *PB* of **Pena_Book** class.
  - Display a message "Invalid Selection".

Sample output screen . [The bold items are inputs entered by admin.]

```
Enter your staff Name :Henry
Enter Job Status [Permanent or Contract or Part-Time] : Permanent

Welcome to Pena Book Salary System.
Enter your transportation allowance :1000
Enter your meal allowance :500
Enter bonus %:5
Staff Name : Henry
Bonus : 80.00
Basic Salary : 1600.00
Net Salary : 3180.00


Thank you.

Enter your staff Name :Rohaiza
Enter Job Status [Permanent or Contract or Part-Time] : Contract

Welcome to Pena Book Salary System.
Enter your working hour : 30
Enter bonus %:2
```

Continued...

```
Enter pay per hour :RM 20
Staff Name : Rohaiza
Bonus : 12.00
Salary :600.00
Net Salary : 612.00

Thank you.

Enter your staff Name :Faiyaz
Enter Job Status [Permanent or Contract or Part-Time] : Part-Time

Welcome to Pena Book Salary System.
Enter your total working hour : 30
Enter bonus %:3
Enter pay per hour :RM 10
Staff Name : Faiyaz
Bonus : 9.00
Salary :300.00
Net Salary : 309.00

Thank you.
```

End of Page.